

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
МИКОЛАЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ В. О. СУХОМЛИНСЬКОГО
Механіко-математичний факультет

Кафедра комп'ютерних наук та прикладної математики




РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
ПРОГРАМУВАННЯ

Ступінь бакалавра


Галузь знань 12 Інформаційні технології
спеціальність 122 «Комп'ютерні науки»
освітня програма Комп'ютерні науки

2019 – 2020 навчальний рік

Розробник: Ходякова Галина Вікторівна, доцент кафедри комп'ютерних наук та прикладної математики, кандидат педагогічних наук  (Ходякова Г.В.)

Робоча програма затверджена на засіданні кафедри комп'ютерних наук та прикладної математики

Протокол № 1 від «27» серпня 2019 р.

Завідувач кафедри  (Поздєєв В.О.)

«27» серпня 2019 р.

1. Опис навчальної дисципліни

Найменування показників	Галузь знань, освітній ступінь	Характеристика навчальної дисципліни	
		<i>денна форма навчання</i>	
Кількість кредитів – 5	Галузь знань 12 Інформаційні технології	Вибіркова	
	Спеціальність 122 Комп'ютерні науки		
Загальна кількість годин 150	Освітня програма: Комп'ютерні науки	<i>Рік підготовки:</i>	
		2-й	
		<i>Семестр</i>	
		3-й	
Тижневих годин для денної форми навчання: аудиторних – 4 самостійної роботи студента – 6	Ступінь бакалавра	16 год.	
		<i>Практичні, семінарські</i>	
		<i>Лабораторні</i>	
		24 год.	
		<i>Самостійна робота</i>	
		110 год.	
Вид контролю: залік			

Мова навчання – українська.

Примітка.

Співвідношення кількості годин аудиторних занять до самостійної та індивідуальної роботи становить: для денної форми навчання – 40 год. – аудиторні заняття, 110 год.– самостійна робота (30% ~ 70%).

2. Мета, завдання навчальної дисципліни та результати навчання

Мета курсу «Програмування» є ознайомлення студентів із основами ООП, поняттями про його базові концепції (інкапсуляція, поліморфізм, наслідування), застосування в подальшому мов програмування, що підтримують методологію ООП (C++, Java, C#). Формування у студентів знань, вмінь і навичок, необхідних для ефективного використання сучасних технологій програмування для розв'язку різних задач та створення прикладного програмного забезпечення загального та професійного спрямування у своїй майбутній професійній, практичній діяльності. У підсумку, вивчення даної дисципліни студент отримує основу для подальшого вивчення усього циклу «комп'ютерних» дисциплін та самостійного оволодіння будь-якими мовами програмування, в основі яких лежить методологія ООП.

Завдання – набуття знань та навичок у сфері розробки програмного забезпечення

- вивчити основні принципи об'єктно-орієнтованої парадигми програмування, як найбільш поширеної і затребуваної в даний час;
- вивчити основні можливості об'єктно-орієнтованої мови програмування C++;
- вивчити основні методи програмування на мові C++;
- отримати навички практичного програмування на мові C++;
- отримати навички візуального програмування.

Студенти повинні знати:

- основні можливості об'єктно-орієнтованої мови і засоби їх використання;
- будову і загальні підходи до використання засобів стандартних бібліотек;
- основні принципи реалізації класів;
- технології об'єктно-орієнтованої розробки програм;
- технології розробки програм на мові C++;

вміти:

- описувати класи, їх атрибути і методи;
- розуміти призначення та використовувати конструктори, деструктори, перевантажені (overloaded) функції та оператори;
- описувати віртуальні (virtual) методи та абстрактні (abstract) класи;
- описувати і використовувати статичні (static) атрибути та методи класів;
- описувати та використовувати дружні (friend) функції класів;
- застосовувати при розробці класів інкапсуляцію, поліморфізм та наслідування (в т.ч. множинне);
- вміти самостійно опановувати нові методи та технології розробки програм.

Передумови для вивчення дисципліни: алгоритми та структури даних, дискретна математика, теорія графів, обчислення матриць.

Навчальна дисципліна складається з 5 кредитів.

Очікувані результати навчання:

Застосовувати знання основних форм і законів абстрактно-логічного мислення, основ методології наукового пізнання, форм і методів вилучення, аналізу, обробки та синтезу інформації в предметній області комп'ютерних наук.

Використовувати методологію системного аналізу об'єктів, процесів і систем для задач аналізу, прогнозування, управління та проектування процесів у прикладних задачах.

I. Загальнопредметні:

ЗК1. Здатність до абстрактного мислення, аналізу та синтезу.

ЗК17 Професійне володіння комп'ютером та інформаційними технологіями

КЗН-4 Базові знання з системних та кібернетичних наук, необхідних для засвоєння загально-професійних дисциплін з інформатики.

- КЗН-5 Базові знання в галузі інформатики й сучасних інформаційних технологій.
- КІ-5 Знання методів та правил роботи з комп'ютером та роботи в Інтернеті.
- КІ-6 Знання законів, методів та методик проведення наукових та прикладних досліджень.
- КЗП-4 Знання вимог чинних державних та міжнародних стандартів, методів і засобів проектування комп'ютеризованих систем, життєвого циклу їх програмного забезпечення.
- КЗП-5 Знання та розуміння основ програмування, мов різних рівнів та їхніх переваг для розв'язання конкретних задач, методів розроблення програмного забезпечення комп'ютеризованих систем з використанням сучасних технологій.

II. Фахові:

- КСП-7 Знання методів виявлення, формулювання, специфікації, аналізу та трасування вимог до комп'ютеризованих систем на етапі їх проектування, методів проектування та верифікації абстрактної архітектури комп'ютеризованих систем.
- КСП-15 Знання операційних систем (Windows, Unix тощо), системного програмного забезпечення, найбільш розповсюджених пакетів прикладних програм, інформаційних порталів Інтернет, програмних методів захисту інформації в комп'ютеризованих системах та мережах
- КСП-16 Знання базових та спеціалізованих технологій розроблення програмного забезпечення комп'ютеризованих систем.
- КСП-17 Знання методів, методик контролю та тестування правильності роботи програмного забезпечення комп'ютеризованих систем.

3. Програма навчальної дисципліни

Кредит 1. Основні принципи ООП. Класи та об'єкти.

Тема 1. Огляд об'єктно-орієнтованого програмування.

Історія програмування. Модель об'єктно-орієнтованого програмування. Абстракція даних. Інкапсуляція. Наслідування. Поліморфізм. Методологія ООП.

Тема 2. Класи.

Синтаксис оголошення класу. Приклад оголошення класу. Визначення функції-члена класу. Створення об'єкта класу. Використання вказівників на об'єкт. Динамічні об'єкти. Використання операції розширення видимості. Обмеження на дані члени-класу. Неповне оголошення класу. Специфікатори доступу. Створення класу в середовищі розробки Code Blocks.

Кредит 2. Елементи класів.

Тема 3. Функції-члени класу.

Конструктори і деструктори. Правила для конструкторів. Правила для деструкторів. Список ініціалізації елементів. Конструктори по замовчуванню. Конструктори копіювання. Вказівник this. Вбудовані функції (inline). Статичні дані-члени класу. Статичні функції-члени класу. Константні дані-члени класу. Константні функції-члени класу. Константні об'єкти.

Тема 4. Дружні функції. Дружні класи.

Функції дружні одному класу. Функції дружні декільком класам. Функції-члени дружні іншому класу.

Кредит 3. Реалізація принципу наслідування у ООП.

Тема 5. Просте наслідування.

Робота конструктора і деструктора. Базовий клас координат. Похідний клас точки. Похідний клас вектора. Використання конструктора з параметрами. Перевизначення і виклик членів базового класу.

Тема 6. Множинне наслідування.

Робота конструктора і деструктора. Приклад множинного наслідування. Віртуальні базові класи.

Кредит 4. Механізм перевантаження у ООП

Тема 7. Перевантаження функцій.

Призначення перевантаження. Декорування компілятором імен функцій. Перевантаження функцій. Перевантаження конструкторів. Неоднозначність перевантаження.

Тема 8. Перевантаження операторів..

Поняття перевантаження операторів. Правила перевантаження операторів. Перевантаження унарних операторів. Перевантаження оператора присвоювання. Перевантаження бінарних операторів. Перевантаження операторів $X=$. Перевантаження бінарних операторів, які використовують об'єкти двох класів. Перевантаження операторів інкременту і декременту. Перевантаження оператора індексування. Перевантаження оператора виклику функції. Перетворення типу.

Кредит 5. Принцип поліморфізму та обробка виключень.

Тема 9. Віртуальні функції.

Раннє і пізнє зв'язування. Динамічний поліморфізм. Віртуальні функції. Віртуальні деструктори. Абстрактні класи і чисто віртуальні функції.

Тема 10. Потокове введення – виведення. Обробка виключень.

Зумовлені потоки. Переваги потоків бібліотеки C++. Операції поміщення і витягання із потоку. Форматування потоку. Файлове введення – виведення з використанням потоків. Обробка виключних ситуацій. Генерування виключень.

4. Структура навчальної дисципліни

Назви змістових модулів і тем	Кількість годин					
	Денна форма					
	Усього	у тому числі				
		л	п	лаб	інд	ср
Кредит 1. Основні принципи ООП. Класи та об'єкти.						
Тема 1. Огляд об'єктно-орієнтованого програмування.	14	2		2		10
Тема 2. Класи.	16	2		4		10
Усього	30	4		6		20
Кредит 2. Елементи класів.						
Тема 3. Функції-члени класу.	14	2		2		10
Тема 4. Дружні функції. Дружні класи.	16	2		4		10
Усього	30	4		6		20
Кредит 3. Реалізація принципу наслідування у ООП.						
Тема 5. Просте наслідування.	16	2		2		12
Тема 6. Множинне наслідування.	14			2		12
Усього	30	2		4		24
Кредит 4. Механізм перевантаження у ООП						
Тема 7. Перевантаження функцій.	14	2		2		10
Тема 8. Перевантаження операторів	16	2		2		12
Усього	30	4		4		22
Кредит 5. Принцип поліморфізму та обробка виключень.						
Тема 9. Віртуальні функції	16	2		2		12
Тема 10. Потокowe введення – виведення. Обробка виключень.	14			2		12
Усього	30	2		4		24
Усього годин:	150	16		24		110

5. Теми лекційних занять

№ з/п	Назва теми	Кількість годин
	<i>Кредит 1. Основні принципи ООП. Класи та об'єкти.</i>	
1	Тема 1. Огляд об'єктно-орієнтованого програмування.	2
2	Тема 2. Класи.	2
	<i>Кредит 2. Елементи класів.</i>	
3	Тема 3. Функції-члени класу.	2
4	Тема 4. Дружні функції. Дружні класи.	2
<i>Кредит 3. Реалізація принципу наслідування у ООП.</i>		
5	Тема 5. Просте наслідування. Множинне наслідування.	2
<i>Кредит 4. Механізм перевантаження у ООП</i>		
6	Тема 6. Перевантаження функцій.	2
7	Тема 7. Перевантаження операторів	2
<i>Кредит 5. Принцип поліморфізму та обробка виключень</i>		
8	Тема 8. Віртуальні функції. Потокowe введення – виведення. Обробка виключень.	2
	Разом:	16

6. Теми лабораторних занять

№ з/п	Назва теми	Кількість годин
	<i>Кредит 1. Основні принципи ООП. Класи та об'єкти.</i>	
1	Тема 1. Створення проектів зі структурами.	2
2	Тема 2. Створення класу.	2
	<i>Кредит 2. Елементи класів.</i>	
3	Тема 3. Класи та дані-члени класів	2
4	Тема 4. Функції-члени класів	2

5	Тема 5. Види конструкторів	2
<i>Кредит 3. Реалізація принципу наслідування у ООП.</i>		
6	Тема 6. Розробка класів та їх структурних елементів	2
7	Тема 7. Масиви у класах. Масиви об'єктів	2
<i>Кредит 4. Механізм перевантаження у ООП</i>		
8	Тема 8. Дружні функції. Розробка різних видів-методів класу	2
9	Тема 9. Просте наслідування. Множинне наслідування.	2
<i>Кредит 5. Принцип поліморфізму та обробка виключень</i>		
10	Тема 10. Перевантаження унарних та бінарних операторів.	2
11	Тема 10. Оператори виведення і вставки в потік.	2
12	Тема 12. Файлове введення і виведення.	2
	Разом:	24

7. Самостійна робота

№ з/п	Назва теми	Кількість годин
<i>Кредит 1. Основні принципи ООП. Класи та об'єкти.</i>		
1.	Тема 1. Опрацювання структур	7
2.	Тема 2. Види конструкторів	7
	Оформлення звітів. Підготовка до захисту робіт	6
<i>Кредит 2. Елементи класів.</i>		
3.	Тема 3. Перевантаження операцій.	8
4.	Тема 4. Абстрактні класи.	7
	Оформлення звітів. Підготовка до захисту робіт	5
<i>Кредит 3. Реалізація принципу наслідування у ООП.</i>		
5.	Тема 5. Шаблони класів.	7
6.	Тема 6. Опрацювання виключних ситуацій.	7

	Оформлення звітів. Підготовка до захисту робіт	8
<i>Кредит 4. Механізм перевантаження у ООП</i>		
7.	Тема 7. Шаблони класів динамічних масивів.	6
8.	Тема 8. Потоківі класи.	6
	Оформлення звітів. Підготовка до захисту робіт	6
<i>Кредит 5. Принцип поліморфізму та обробка виключень.</i>		
9.	Тема 9. Спеціальні класи, функції і вказівники.	7
10.	Тема 10. Препроцесор і компілятор.	7
	Оформлення звітів. Підготовка до захисту робіт	6
Всього :		110

8. Форми роботи та критерії оцінювання

Рейтинговий контроль знань студентів здійснюється за 100-бальною шкалою.

Шкала оцінювання: національна та ECTS

ОЦІНКА ЄКТС	СУМА БАЛІВ	ОЦІНКА ЗА НАЦІОНАЛЬНОЮ ШКАЛОЮ	
		екзамен	залік
A	90-100	5 (відмінно)	5/відм./зараховано
B	80-89	4 (добре)	4/добре/ зараховано
C	65-79		
D	55-64	3 (задовільно)	3/задов./ зараховано
E	50-54		
FX	35-49	2 (незадовільно)	Не зараховано

Форми поточного та підсумкового контролю. Комплексна діагностика знань, умінь і навичок студентів із дисципліни здійснюється на основі результатів проведення поточного й підсумкового контролю знань (КР). Поточне оцінювання (індивідуальне, групове і фронтальне опитування, самостійна робота, самоконтроль). Завданням поточного контролю є систематична перевірка розуміння та засвоєння програмового матеріалу, виконання практичних, лабораторних робіт, уміння самостійно опрацьовувати тексти, складання конспекту рекомендованої літератури, написання і захист реферату, здатності публічно чи письмово представляти певний матеріал.

Завданням підсумкового контролю (КР, залік) є перевірка глибини засвоєння студентом програмового матеріалу модуля.

Критерії оцінювання відповідей на практичних заняттях:

Студенту виставляється відмінно, якщо студент здатний самостійно здійснювати основні види навчальної діяльності. Знання студента є глибокими, міцними, узагальненими;

студент вміє застосовувати знання творчо, його навчальна діяльність позначена вмінням самостійно оцінювати різноманітні життєві ситуації, явища, факти, виявляти і відстоювати особисту позицію.

Студенту виставляється дуже добре, якщо студент знає істотні ознаки понять, явищ, закономірностей, зв'язків між ними, а також самостійно застосовує знання в нестандартних ситуаціях, володіє розумовими операціями, вміє робити висновки, виправляти допущені помилки. Відповідь повна, правильна, логічна, обґрунтована.

Студенту виставляється добре, якщо студент знає ознаки понять, явищ, закономірностей, зв'язків між ними на середньому рівні, а також самостійно застосовує знання в стандартних ситуаціях, володіє розумовими операціями, вміє робити висновки, виправляти допущені помилки. Відповідь повна, правильна, логічна, обґрунтована.

Студенту виставляється достатньо, якщо відповідь студента при відтворенні навчального матеріалу елементарна, зумовлюється початковими уявленнями про предмет вивчення. Студент відтворює основний навчальний матеріал, здатний виконувати завдання за зразком, володіє елементарними вміннями навчальної діяльності.

Студенту виставляється мінімальний задовільно, якщо відповідь студента при відтворенні навчального матеріалу елементарна, зумовлюється початковими уявленнями про предмет вивчення. Студент відтворює основний навчальний матеріал.

Кількість балів у кінці семестру повинна складати від 250 до 500 балів (за 5 кредитів), тобто сума балів за виконання усіх завдань.

Відповідний розподіл балів, які отримують студенти за 5 крд

Поточне оцінювання та самостійна робота										Контр робота	Накоп. бали/ Сума
T1	T2	T3	T4	T5	T6	T7	T8	T9	T10		
40	40	40	40	40	40	40	40	40	40	100	500/100

9. Засоби діагностики

Засобами діагностики та методами демонстрування результатів навчання є: завдання до лабораторних занять, завдання для самостійної та індивідуальної роботи, презентації результатів досліджень, тестові завдання, контрольні роботи.

10. Методи навчання

Курс складається з лекційних, лабораторних занять та самостійної роботи студентів, домашніх завдань і завершується підсумковим рейтинг-контролем і виконанням конкретних контрольних завдань по даній дисципліні.

Лекційні заняття призначені для теоретичного осмислення і узагальнення базових розділів курсу.

Лабораторні заняття є аудиторними, проводяться по наперед відомих темах у вигляді практичної роботи студентів за комп'ютерами. Лабораторне заняття – форма навчального заняття, спрямована на закріплення та вдосконалення студентом теоретичних знань, отриманих як на лекційних заняттях, так і в процесі самостійного вивчення матеріалу. Під час лабораторного заняття студенти під керівництвом викладача особисто набувають

практичних навичок у роботі з обчислювальною технікою, оволодівають методикою створення програмних продуктів у програмному середовищі. У ході заняття студенти самостійно виконують передбачені завданням дії, заносючи результати в звіт. На це відводиться до 85 – 90% часу заняття. Викладач здійснює контроль за роботою і надає допомогу при виникненні труднощів, звертає увагу на складні, ключові моменти. Причому основна увага приділяється не вказівці на конкретну помилку, а методиці пошуку причин виникнення цих помилок. Складання звіту – це відповідальний етап лабораторного заняття. При його складанні студенти розвивають навички аналізу, узагальнення і творчого осмислення результатів роботи, а також навички розробки документації. Кожна лабораторна робота передбачає наявність індивідуального завдання і наступного захисту.

Самостійна робота є позааудиторною і призначена для самостійного ознайомлення студента з певними розділами курсу за рекомендованими педагогом матеріалами і підготовки до виконання індивідуальних завдань по курсу.

Поточний рейтинг-контроль проводиться викладачем в процесі проведення всіх видів занять. Проміжний рейтинг-контроль призначений для практичної комплексної оцінки освоєння розділів курсу і здійснюється шляхом підготовки студентами відповідей на поставлені питання.

Кінцевий контроль знань з дисципліни проводиться під час складання іспиту.

11. Методичне забезпечення

Бібліотека університету має достатню кількість підручників та посібників, лабораторних практикумів та іншої літератури для забезпечення повноцінного навчання студентів.

Крім цього на кафедрі наявна бібліотека для користування студентами на електронних носіях.

Конспекти лекцій, інструкції до лабораторних робіт та додаткові матеріали розміщуються на сервері кафедри та у навчально-методичному комплексі дисципліни.

12. Рекомендована література

Базова

1. Александреску А. - Современное проектирование на C++ - М: Издательский дом «Вильямс», 2002 – 336 с.
2. Астахова И.Ф., Власов С.В. - Язык C++. Учебное пособие – Мн: Новое название, 2003 – 203 с. Роберт Лафоре. Объектно-ориентированное программирование в C++ - Питер, 2013 – 928 с.
3. Страуступ, Б. Язык программирования C++. Специальное издание / Б. Страуступ. - М.: Бином, 2015. - 1136 с.
4. Бондарев В.М.- Программирование на C++ - Харьков: «Компания СМІТ», 2005 – 284 с.

Допоміжна

1. Э. Гамма, Р. Хелм, Р. Джонсон, Д. Влссидес: Приемы ООП. Паттерны проектирования – Питер, 2001 – 368 с.

2. В.В. Мухортов, В.Ю. Рылов: Объектно-Ориентированное Программирование, Анализ и Дизайн – Новосибирск, 2006 – 103 с.
3. Тимоти Бадд: Объектно-ориентированное программирование в действии – 2006
4. Хэнкон Л., Кригер М. Введение в программирование на языке Си. - М.: Сов. радио, 1986.
5. Белецкий Я. Энциклопедия языка Си. – М.: Мир, 1992.
6. С++. Язык программирования. – М.: “И.В.К.-СОФТ”, 1991.
7. Березин В.Н., Березин И.В. Начальный курс Си и Си++. – М.: Диалог-Мифи, 1996.
8. Иан Грэхем Объектно-ориентированные методы. Принципы и практика = Object-Oriented Methods: Principles & Practice. — 3-е изд. — М.: «Вильямс», 2004. — С. 880. — ISBN 0-201-61913-X
9. Антони Синтес Освой самостоятельно объектно-ориентированное программирование за 21 день = Sams Teach Yourself Object-Oriented Programming in 21 Days. — М.: «Вильямс», 2002. — С. 672. — ISBN 0-672-32109-2
10. Ашарина, И.В. Основы программирования на языках С и С++ / И.В. Ашарина. - М.: ГЛТ, 2012. - 208 с.
11. Вирт Н. Алгоритмы и структуры данных. - М.: Мир, 1989. - 360 с.
12. Ахо А., Хопкрофт Д., Ульман Д. Структуры данных и алгоритмы. М.: Вильямс, 2000
13. М. Эллис, Б. Страуструп. Справочное руководство по языку С++ с комментариями: Пер. с англ. - Москва: Мир, 1992. 445с.
14. Стенли Б. Липшман. С++ для начинающих: Пер. с англ. 2тт. - Москва: Унитех; Рязань: Гэлион, 1992, 304-345сс.
15. Бруно Бабэ. Просто и ясно о Borland С++: Пер. с англ. - Москва: БИНОМ, 1994. 400с.
16. В. Липский Комбинаторика для программистов.-1988. 200 с.
17. С.Окулов Программирование в алгоритмах.- 2006. 220с.
18. Серджвик Р. Фундаментальные алгоритмы на С.Часть 1-3 СПб: Диасофт, 2003.
19. Кениг, Э. Эффективное программирование на С++. Практическое программирование на примерах. Т. 2 / Э. Кениг, Б.Э. Му. - М.: Вильямс, 2016. - 368 с.
20. МакГрат, М. Программирование на С для начинающих / М. МакГрат. - М.: Эксмо, 2015. - 192 с.
21. Мартынов, Н.Н. Программирование для Windows на С / Н.Н. Мартынов. - М.: БИНОМ, 2013. - 528 с.
22. Перри, Г. Программирование на С для начинающих / Г. Перри, Д. Миллер. - М.: Эксмо, 2015. - 368 с.
23. 9. Фленов, М.Е. Программирование на С++ глазами хакера. / М.Е. Фленов. - СПб.: ВHV, 2012. - 352 с.
24. Хенкеманс, Д. Программирование на С++ / Д. Хенкеманс, М. Ли. - СПб.: Символ-плюс, 2015. - 416 с.
25. Бочков С.О. Субботин Д.М. Язык программирования С для персонального компьютера. - М.: Радио и связь, 1990.
26. В.В. Подбельский. Язык С++: Учебное пособие. - Москва: Финансы и статистика, 1995. 560с.
27. Г. Шилдт. Самоучитель С++: Пер. с англ. - Санкт-Петербург: ВHV-Санкт-Петербург, 1998. 620с.
28. У. Сэвитч. С++ в примерах: Пер. с англ. - Москва: ЭКОМ, 1997. 736с.
29. К. Джемса. Учимся программировать на языке С++: Пер. с англ. - Москва: Мир, 1997. 320с.
30. Керниган Б., Ритчи Д. Язык программирования С. - М.: Финансы и статистика, 1985.
31. Керниган Б., Ритчи Д., Фьюэр А. Язык программирования С. Задачи по языку С. - М.: Финансы и статистика, 1985.
32. Кормен Т.,Лейзерсон Ч., Ривест Р., Штайн К. Алгоритмы: построение и анализ, 2-е издание – М.: Вильямс, 2006.

33. Абельсон Х., Сассман Д. Структура и интерпретация компьютерных программ. М.: Добросвет, 2006
34. Уильям Топп, Уильям Форд. Структуры данных в С++: Пер. с англ. – М.: ЗАО «Издательство БИНОМ», 2000. – 816 с.
35. Караванова Т.П. Информатика. Методи побудови алгоритмів та їх аналіз. - К.:Генеза. 2007. 216 с.
36. Дорогов, В.Г. Основы программирования на языке С: Учебное пособие / В.Г. Дорогов, Е.Г. Дорогова; Под общ. ред. проф. Л.Г. Гагарина. - М.: ИД ФОРУМ, НИЦ ИНФРА-М, 2013. - 224 с.

15. Інформаційні ресурси

1. http://ru.wikipedia.org/wiki/Объектно-ориентированное_программирование
2. <http://www.youtube.com/watch?v=rbCkcBDobCY>
3. <http://www.codenet.ru/progr/vbasic/oop.php>
4. <http://sdb.su/oop/>
5. <http://younglinux.info/oopython.php>
6. <http://msdn.microsoft.com/ru-ru/library/dd460654.aspx>
7. <http://www.weblibrary.biz/c-sharp/principy>
8. <http://www.myoop.ru/>
9. <http://phpjs.ru/material-62>
10. <http://proger.elitno.net/oop/121-preimushchestva-i-nedostatki-oop-obektno-orientirovannye-yazyki-programmirovaniya>

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
МИКОЛАЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ В. О. СУХОМЛИНСЬКОГО**

Механіко-математичний факультет

Кафедра комп'ютерних наук та прикладної математики



**ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
ПРОГРАМУВАННЯ**

Ступінь бакалавра

Галузь знань 12 Інформаційні технології
спеціальність 122 «Комп'ютерні науки»
освітня програма Комп'ютерні науки

Миколаїв – 2019

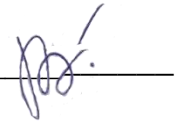
Програму розроблено та внесено: Миколаївський національний університет імені В. О. Сухомлинського

РОЗРОБНИК ПРОГРАМИ: Ходякова Галина Вікторівна, доцент кафедри комп'ютерних наук та прикладної математики, кандидат педагогічних наук.

Програму схвалено на засіданні кафедри комп'ютерних наук та прикладної математики

Протокол від «27» серпня 2019 року № 1

Завідувач кафедри комп'ютерних наук та прикладної математики (Поздєєв В.О.)



Програму погоджено навчально-методичною комісією механіко-математичного факультету

Протокол від «27» серпня 2019 року №

Голова навчально-методичної комісії




(Пархоменко О.Ю.)

Програму погоджено навчально-методичною комісією університету

Протокол від «27» серпня 2019 року № 14

Голова навчально-методичної комісії університету



(Кузнецова О.А.)

Програма вивчення нормативної вибіркової дисципліни «Програмування» складена відповідно до освітнього ступеня підготовки бакалаврів спеціальності 122 Комп'ютерні науки.

Предметом вивчення навчальної дисципліни є об'єктно-орієнтоване програмування на C++.

1. Мета та завдання навчальної дисципліни

Мета курсу «Програмування» є ознайомлення студентів із основами ООП, поняттями про його базові концепції (інкапсуляція, поліморфізм, наслідування), застосування в подальшому мов програмування, що підтримують методологію ООП (C++, Java, C#). Формування у студентів знань, вмінь і навичок, необхідних для ефективного використання сучасних технологій програмування для розв'язку різних задач та створення прикладного програмного забезпечення загального та професійного спрямування у своїй майбутній професійній, практичній діяльності. У підсумку, вивчення даної дисципліни студент отримує основу для подальшого вивчення усього циклу «комп'ютерних» дисциплін та самостійного оволодіння будь-якими мовами програмування, в основі яких лежить методологія ООП.

Завдання – набуття знань та навичок у сфері розробки програмного забезпечення

- вивчити основні принципи об'єктно-орієнтованої парадигми програмування, як найбільш поширеної і затребуваної в даний час;
- вивчити основні можливості об'єктно-орієнтованої мови програмування C++;
- вивчити основні методи програмування на мові C++;
- отримати навички практичного програмування на мові C++;
- отримати навички візуального програмування.

Студенти повинні знати:

- основні можливості об'єктно-орієнтованої мови і засоби їх використання;
- будову і загальні підходи до використання засобів стандартних бібліотек;
- основні принципи реалізації класів;
- технології об'єктно-орієнтованої розробки програм;
- технології розробки програм на мові C++;

вміти:

- описувати класи, їх атрибути і методи;
- розуміти призначення та використовувати конструктори, деструктори, перевантажені (overloaded) функції та оператори;
- описувати віртуальні (virtual) методи та абстрактні (abstract) класи;
- описувати і використовувати статичні (static) атрибути та методи класів;
- описувати та використовувати дружні (friend) функції класів;
- застосовувати при розробці класів інкапсуляцію, поліморфізм та наслідування (в т.ч. множинне);
- вміти самостійно опановувати нові методи та технології розробки програм.

Передумови для вивчення дисципліни: алгоритми та структури даних, дискретна математика, теорія графів, обчислення матриць.

Очікувані результати навчання:

Застосовувати знання основних форм і законів абстрактно-логічного мислення, основ методології наукового пізнання, форм і методів вилучення, аналізу, обробки та синтезу інформації в предметній області комп'ютерних наук.

Використовувати методологію системного аналізу об'єктів, процесів і систем для задач аналізу, прогнозування, управління та проектування процесів у прикладних задачах.

I. Загальнопредметні:

ЗК1. Здатність до абстрактного мислення, аналізу та синтезу.

ЗК17 Професійне володіння комп'ютером та інформаційними технологіями

КЗН-4 Базові знання з системних та кібернетичних наук, необхідних для засвоєння загально-професійних дисциплін з інформатики.

- КЗН-5 Базові знання в галузі інформатики й сучасних інформаційних технологій.
- КІ-5 Знання методів та правил роботи з комп'ютером та роботи в Інтернеті.
- КІ-6 Знання законів, методів та методик проведення наукових та прикладних досліджень.
- КЗП-4 Знання вимог чинних державних та міжнародних стандартів, методів і засобів проектування комп'ютеризованих систем, життєвого циклу їх програмного забезпечення.
- КЗП-5 Знання та розуміння основ програмування, мов різних рівнів та їхніх переваг для розв'язання конкретних задач, методів розроблення програмного забезпечення комп'ютеризованих систем з використанням сучасних технологій.

II. Фахові:

- КСП-7 Знання методів виявлення, формулювання, специфікації, аналізу та трасування вимог до комп'ютеризованих систем на етапі їх проектування, методів проектування та верифікації абстрактної архітектури комп'ютеризованих систем.
- КСП-15 Знання операційних систем (Windows, Unix тощо), системного програмного забезпечення, найбільш розповсюджених пакетів прикладних програм, інформаційних порталів Інтернет, програмних методів захисту інформації в комп'ютеризованих системах та мережах
- КСП-16 Знання базових та спеціалізованих технологій розроблення програмного забезпечення комп'ютеризованих систем.
- КСП-17 Знання методів, методик контролю та тестування правильності роботи програмного забезпечення комп'ютеризованих систем.

2. Інформаційний обсяг навчальної дисципліни

Кредит 1. Основні принципи ООП. Класи та об'єкти.

Тема 1. Огляд об'єктно-орієнтованого програмування.

Історія програмування. Модель об'єктно-орієнтованого програмування. Абстракція даних. Інкапсуляція. Наслідування. Поліморфізм. Методологія ООП.

Тема 2. Класи.

Синтаксис оголошення класу. Приклад оголошення класу. Визначення функції-члена класу. Створення об'єкта класу. Використання вказівників на об'єкт. Динамічні об'єкти. Використання операції розширення видимості. Обмеження на дані члени-класу. Неповне оголошення класу. Специфікатори доступу. Створення класу в середовищі розробки Code Blocks.

Кредит 2. Елементи класів.

Тема 3. Функції-члени класу.

Конструктори і деструктори. Правила для конструкторів. Правила для деструкторів. Список ініціалізації елементів. Конструктори по замовчуванню. Конструктори копіювання. Вказівник this. Вбудовані функції (inline). Статичні дані-члени класу. Статичні функції-члени класу. Константні дані-члени класу. Константні функції-члени класу. Константні об'єкти.

Тема 4. Дружні функції. Дружні класи.

Функції дружні одному класу. Функції дружні декільком класам. Функції-члени дружні іншому класу.

Кредит 3. Реалізація принципу наслідування у ООП.

Тема 5. Просте наслідування.

Робота конструктора і деструктора. Базовий клас координат. Похідний клас точки. Похідний клас вектора. Використання конструктора з параметрами. Перевизначення і виклик членів базового класу.

Тема 6. Множинне наслідування.

Робота конструктора і деструктора. Приклад множинного наслідування. Віртуальні базові класи.

Кредит 4. Механізм перевантаження у ООП

Тема 7. Перевантаження функцій.

Призначення перевантаження. Декорування компілятором імен функцій. Перевантаження функцій. Перевантаження конструкторів. Неоднозначність перевантаження.

Тема 8. Перевантаження операторів..

Поняття перевантаження операторів. Правила перевантаження операторів. Перевантаження унарних операторів. Перевантаження оператора присвоювання. Перевантаження бінарних операторів. Перевантаження операторів $X=$. Перевантаження бінарних операторів, які використовують об'єкти двох класів. Перевантаження операторів інкременту і декременту. Перевантаження оператора індексування. Перевантаження оператора виклику функції. Перетворення типу.

Кредит 5. Принцип поліморфізму та обробка виключень.

Тема 9. Віртуальні функції.

Раннє і пізнє зв'язування. Динамічний поліморфізм. Віртуальні функції. Віртуальні деструктори. Абстрактні класи і чисто віртуальні функції.

Тема 10. Потокове введення – виведення. Обробка виключень.

Зумовлені потоки. Переваги потоків бібліотеки C++. Операції поміщення і витягання із потоку. Форматування потоку. Файлове введення – виведення з використанням потоків. Обробка виключних ситуацій. Генерування виключень.

3. Рекомендована література

Базова

1. Роберт Лафоре. Объектно-ориентированное программирование в C++ - Питер, 2013 – 928 с.
2. Бондарев В.М.- Программирование на C++ - Харьков: «Компания СМИТ», 2005 – 284 с.
3. В.В. Мухортов, В.Ю. Рылов: Объектно-Ориентированное Программирование, Анализ и Дизайн – Новосибирск, 2006 – 103 с.
4. Тимоти Бадд: Объектно-ориентированное программирование в действии – 2006
5. Астахова И.Ф., Власов С.В. - Язык C++. Учебное пособие – Мн: Новое название, 2003 – 203 с.
6. Александреску А. - Современное проектирование на C++ - М: Издательский дом «Вильямс», 2002 – 336 с.

Допоміжна

1. Э. Гамма, Р. Хелм, Р. Джонсон, Д. Влассидес: Приемы ООП. Паттерны проектирования – Питер, 2001 – 368 с.
2. Астахова И.Ф., Власов С.В. - Язык C++. Учебное пособие – Мн: Новое название, 2003 – 203 с.
3. Хэнкон Л., Кригер М. Введение в программирование на языке Си. - М.: Сов. радио, 1986.
4. Белецкий Я. Энциклопедия языка Си. – М.: Мир, 1992.
5. C++. Язык программирования. – М.: “И.В.К.-СОФТ”, 1991.
6. Березин В.Н., Березин И.В. Начальный курс Си и Си++. – М.: Диалог-Мифи, 1996.

7. Иан Грэхем Объектно-ориентированные методы. Принципы и практика = Object-Oriented Methods: Principles & Practice. — 3-е изд. — М.: «Вильямс», 2004. — С. 880. — ISBN 0-201-61913-X
8. Антони Синтес Освой самостоятельно объектно-ориентированное программирование за 21 день = Sams Teach Yourself Object-Oriented Programming in 21 Days. — М.: «Вильямс», 2002. — С. 672. — ISBN 0-672-32109-2
9. Ашарина, И.В. Основы программирования на языках С и С++ / И.В. Ашарина. - М.: ГЛТ, 2012. - 208 с.
10. Вирт Н. Алгоритмы и структуры данных. - М.: Мир, 1989. - 360 с.
11. Ахо А., Хопкрофт Д., Ульман Д. Структуры данных и алгоритмы. М.: Вильямс, 2000
12. М. Эллис, Б. Страуструп. Справочное руководство по языку С++ с комментариями: Пер. с англ. - Москва: Мир, 1992. 445с.
13. Стенли Б. Липпман. С++ для начинающих: Пер. с англ. 2тт. - Москва: Унитех; Рязань: Гэлион, 1992, 304-345сс.
14. Бруно Бабэ. Просто и ясно о Borland С++: Пер. с англ. - Москва: БИНОМ, 1994. 400с.
15. В. Липский Комбинаторика для программистов.-1988. 200 с.
16. С.Окулов Программирование в алгоритмах.- 2006. 220с.
17. Серджвик Р. Фундаментальные алгоритмы на С.Часть 1-3 СПб: Диасофт, 2003.
18. Кениг, Э. Эффективное программирование на С++. Практическое программирование на примерах. Т. 2 / Э. Кениг, Б.Э. Му. - М.: Вильямс, 2016. - 368 с.
19. МакГрат, М. Программирование на С для начинающих / М. МакГрат. - М.: Эксмо, 2015. - 192 с.
20. Мартынов, Н.Н. Программирование для Windows на С / Н.Н. Мартынов. - М.: БИНОМ, 2013. - 528 с.
21. Перри, Г. Программирование на С для начинающих / Г. Перри, Д. Миллер. - М.: Эксмо, 2015. - 368 с.
22. 9. Фленов, М.Е. Программирование на С++ глазами хакера. / М.Е. Фленов. - СПб.: ВHV, 2012. - 352 с.
23. Хенкеманс, Д. Программирование на С++ / Д. Хенкеманс, М. Ли. - СПб.: Символ-плюс, 2015. - 416 с.
24. Бочков С.О. Субботин Д.М. Язык программирования С для персонального компьютера. - М.: Радио и связь, 1990.
25. В.В. Подбельский. Язык С++: Учебное пособие. - Москва: Финансы и статистика, 1995. 560с.
26. Г. Шилдт. Самоучитель С++: Пер. с англ. - Санкт-Петербург: ВHV-Санкт-Петербург, 1998. 620с.
27. У. Сэвитч. С++ в примерах: Пер. с англ. - Москва: ЭКОМ, 1997. 736с.
28. К. Джамса. Учимся программировать на языке С++: Пер. с англ. - Москва: Мир, 1997. 320с.
29. Керниган Б., Ритчи Д. Язык программирования С. - М.: Финансы и статистика, 1985.
30. Керниган Б., Ритчи Д., Фьюэр А. Язык программирования С. Задачи по языку С. - М.: Финансы и статистика, 1985.
31. Кормен Т.,Лейзерсон Ч., Ривест Р., Штайн К. Алгоритмы: построение и анализ, 2-е издание – М.: Вильямс, 2006.
32. Абельсон Х., Сассман Д. Структура и интерпретация компьютерных программ. М.: Добросвет, 2006
33. Уильям Топп, Уильям Форд. Структуры данных в С++: Пер. с англ. Уильям Топп, Уильям Форд. Структуры данных в С++: Пер. с англ. – М.: ЗАО «Издательство БИНОМ», 2000. – 816 с.
34. Караванова Т.П. Информатика. Методи побудови алгоритмів та їх аналіз. - К.:Генеза. 2007. 216 с.

35. Дорогов, В.Г. Основы программирования на языке С: Учебное пособие / В.Г. Дорогов, Е.Г. Дорогова; Под общ. ред. проф. Л.Г. Гагарина. - М.: ИД ФОРУМ, НИЦ ИНФРА-М, 2013. - 224 с.

Інформаційні ресурси

1. http://ru.wikipedia.org/wiki/Объектно-ориентированное_программирование
2. <http://www.youtube.com/watch?v=rbCkcBDobCY>
3. <http://www.codenet.ru/progr/vbasic/oop.php>
4. <http://sdb.su/oop/>
5. <http://younglinux.info/oopython.php>
6. <http://msdn.microsoft.com/ru-ru/library/dd460654.aspx>
7. <http://www.weblibrary.biz/c-sharp/princypy>
8. <http://www.myoop.ru/>
9. <http://phpjs.ru/material-62>
10. <http://proger.elitno.net/oop/121-preimushchestva-i-nedostatki-oop-obektno-orientirovannye-yazyki-programmirovaniya>

4. Форма підсумкового контролю успішності навчання: іспит

Засоби діагностики успішності навчання:

Лекційні та лабораторні заняття з використанням традиційних методів, інтерактивних методик та комп'ютерної техніки, індивідуальна робота, самостійна робота.

Поточний та підсумковий контроль здійснюється у вигляді комп'ютерних тестів на освітньому просторі університету. Для оцінювання використовується національна чотирьохбальна шкала: відмінно, добре, задовільно, незадовільно; європейська шкала: А, В, С, D, E, FX, F.

(приклад для заліку) 100% балів студенти накопичують на заняттях та під час поточного і підсумкового контролю, що регламентується робочою програмою викладача. (приклад для іспиту) 60% балів студенти накопичують на заняттях та під час поточного контролю, що регламентується робочою програмою викладача, 40% балів студенти набирають на заліку.